Puissance de l'attente aux stations pour l'exploration des réseaux de transport public†

David Ilcinkas et Ahmed Mouhamadou Wade

LaBRI, CNRS & Université de Bordeaux, France

Nous étudions le problème de l'exploration, par une entité mobile, d'une classe de graphes dynamiques appelés graphes périodiquement variables (PV-graphes). Ils sont définis par un ensemble de transporteurs suivant infiniment leur route respective le long des stations du réseau, et modélisent donc naturellement les réseaux de transport public.

Flocchini, Mans et Santoro [FMS09] ont étudié ce problème dans le cas où l'agent doit toujours rester sur les transporteurs. Dans ce papier, nous étudions l'impact de la capacité d'attendre sur les stations. Nous prouvons que l'attente sur les stations permet à l'agent d'atteindre de meilleures complexités en pire cas : le nombre de mouvements est réduit d'un facteur multiplicatif d'au moins $\Theta(p)$, et la complexité en temps passe de $\Theta(kp^2)$ à $\Theta(np)$, où n est le nombre de stations, k le nombre de transporteurs, et p la période maximale ($n \leq kp$ dans tout PV-graphe connexe). Par ailleurs, l'algorithme que nous proposons pour prouver les bornes supérieures permet de réaliser la cartographie du PV-graphe, en plus de l'explorer.

Keywords: Exploration, Graphes dynamiques, Agent mobile, PV-graphes

1 Introduction

1.1 Le problème

Le problème de l'exploration de graphes consiste, pour une entité mobile (aussi appelée agent), à explorer tous les sommets (ou toutes les arrêtes) d'un graphe a priori inconnu. Ce problème étant l'un des plus classiques dans le cadre du calcul par agent mobile, il a reçu beaucoup d'intérêt jusqu'ici. La complexité en temps, en espace, ou l'impact d'une connaissance a priori ont été largement étudiés au cours des 40 dernières années (voir par exemple, [DP04, PP99, Rei05]). Cependant, la grande majorité de ces travaux concerne les graphes statiques, tandis que les nouvelles générations d'environnements interconnectés tendent à être extrêmement dynamiques. Dans ce papier, nous étudions le problème de l'exploration de graphes dans un modèle de réseaux dynamiques, à savoir le modèle des graphes périodiquement variables ou PV-graphes. (Le lecteur intéressé trouvera dans [CFQS10] un récapitulatif très complet des différents modèles de graphes dynamiques existants.)

Grossièrement, un PV-graphe consiste en un ensemble de transporteurs suivant périodiquement leur route respective parmi les sites (sommets) du système. Les PV-graphes modélisent en particulier divers types de systèmes de transport public, par exemple les systèmes de bus ou de métro. Ils modélisent également les systèmes de satellites en orbite autour de la Terre, ou les systèmes de sécurité composé d'agents de sécurité faisant la ronde. Explorer ces environnements permet par exemple d'effectuer une opération de maintenance etc. En effet, un agent peut vérifier si tout est en ordre au cours de l'exploration. Cet agent peut être un logiciel, un robot ou un être humain.

Le problème de l'exploration de PV-graphes a déjà été étudié par Flocchini, Mans et Santoro [FMS09]. Ils ont considéré que l'agent ne peut pas quitter un transporteur pour rester sur un site. Ne pas être capable de rester sur un site est particulièrement légitime dans les systèmes de satellites en orbite autour de la Terre par exemple, où les sites ne correspondent à aucune station physique. Cependant, dans la plupart des

† Avec le soutien du projet ANR ALADDIN, du projet INRIA CEPAGE, et du projet européen EULER. Une version étendue de ces travaux a été publiée dans les actes de la conférence OPODIS 2011 [IW11].

systèmes de transport public, il est possible pour l'agent (humain ou non) de rester sur un site afin d'attendre un transporteur (éventuellement différent). Dans ce papier, nous considérons le même problème mais dans le cas où l'agent peut quitter les transporteurs pour attendre sur un site. Nous étudions l'impact de cette nouvelle capacité sur la complexité (temps et nombre de mouvements) du problème de l'exploration de PV-graphes.

1.2 Précédents résultats

En 2009, Flocchini, Mans et Santoro [FMS09] introduisent un nouveau modèle de graphes dynamiques, le modèle des PV-graphes. Les auteurs montrent que si les sites du PV-graphe sont étiquetés, la connaissance du nombre de sites ou d'une borne supérieure sur la plus grande période est nécessaire pour qu'un agent puisse explorer le PV-graphe. Si les sites du PV-graphe sont anonymes, la connaissance d'une borne supérieure sur la plus grande période est nécessaire. Dans les deux cas, ils prouvent que la complexité (en pire cas) en temps et en mouvements de l'agent est $\Theta(kp^2)$, où k est le nombre de transporteurs et p la période maximale des transporteurs.

1.3 Nos contributions

Dans cet article, nous étendons le travail sur les PV-graphes de Flocchini, Mans et Santoro [FMS09] dans le cas où l'agent peut descendre d'un transporteur et rester sur un site. Cette nouvelle capacité permet à l'agent d'explorer des PV-graphes moins connectés dans le temps (des définitions formelles sont données dans la Section 2). Nous prouvons que, dans le cas général (donc même en considérant les PV-graphes qui ne sont pas hautement connexes), la complexité en mouvements est réduite à $\Theta(min\{kp, np, n^2\})$, et la complexité en temps à $\Theta(np)$. (Notons que dans les PV-graphes connexes, nous avons $n \leq kp$). Nous prouvons également que les complexités en mouvements et en temps dans le cas des PV-graphes hautement connexes restent les mêmes que dans le cas général. Par ailleurs, il se trouve que notre algorithme effectue non seulement l'exploration, mais réalise également la cartographie, c'est à dire qu'il peut produire une copie isomorphe du PV-graphe. Enfin, notons que notre algorithme n'utilise pas les identifiants des sites, tandis que nos bornes inférieures restent vraies dans le cas où l'agent a accès à des identifiants de sites uniques. (Pour des résultats et des preuves plus complets, se référer à la version étendue [IW11].)

2 Définitions et modèles

Nous considérons un système de n sommets (sites) $\{s_1, \ldots, s_n\}$ parcouru par k *transporteurs*. Chaque transporteur c a un identifiant Id(c) et parcourt une séquence ordonnée $R(c) = (s_{i_1}, \cdots, s_{i_{p(c)}})$ de sites, appelée *route*, de manière périodique. L'entier positif $p(c)$ est appelé *période* du transporteur c. Plus précisément, le transporteur c commence sur le site s_{i_1} au temps 0, et parcourt sa route en se déplaçant sur le site suivant à chaque unité de temps de manière cyclique (quand c est sur le site s_{i_p}, il revient au site s_{i_1}).

Un PV-graphe (graphe périodiquement variable) est l'ensemble (S, C), où S est l'ensemble des sites, et C l'ensemble des transporteurs qui circulent sur les sites. Nous notons par n, k et p, respectivement, le nombre de sites, le nombre de transporteurs et le maximum sur les périodes des transporteurs.

Pour tout PV-graphe G, nous définissons deux graphes (classiques) $H_1(G)$ et $H_2(G)$ comme suit. Les deux graphes ont l'ensemble des transporteurs comme ensemble de sommets. Il existe une arête entre deux transporteurs c et c' dans $H_1(G)$ si et seulement si il existe un site qui apparaît dans les routes de c et de c'. Il existe une arête entre deux transporteurs c et c' dans $H_2(G)$ si et seulement si il existe un site s et un temps $t \geq 0$ tel que c et c' se rencontrent sur s au temps t. Un PV-graphe G est dit *connexe* si et seulement si $H_1(G)$ est connexe. Un PV-graphe est dit *hautement connexe* si et seulement si $H_2(G)$ est connexe. Dans cet article, nous considérons les PV-graphes qui sont au moins connexes, les PV-graphes non connexes ne pouvant pas être explorés. Notons que dans le cas où l'agent ne peut pas descendre des transporteurs sur un site, seuls les PV-graphes hautement connexes peuvent être explorés par l'agent.

Une entité externe, appelée *agent*, opère sur ces PV-graphes. Cette entité peut voir un transporteur et connaître son identifiant. Elle peut monter avec un transporteur ou changer de transporteur. Contrairement au modèle utilisé dans [FMS09], l'agent a la possibilité de quitter un transporteur et de rester sur le site courant. Il pourra plus tard remonter avec le même transporteur ou un autre. Nous ne faisons aucune restriction sur la taille de la mémoire de l'agent ou sur ses capacités de calcul. Nous considérons deux modèles concernant les identités des sites. Dans le cas des PV-graphes *anonymes*, les sites n'ont pas d'identifiant, ou l'agent ne peut pas les voir. Dans le cas des PV-graphes *étiquetés*, les sites ont des identifiants distincts et l'agent peut les voir et les mémoriser.

Nous disons qu'un agent explore un PV-graphe si et seulement si, en commençant au temps 0 sur le site de départ du premier transporteur (sans perte de généralité), l'agent peut visiter l'ensemble des sites du PV-graphe en un temps fini et se mettre dans un état terminal. Cet état terminal exprime le fait que l'agent détecte que l'exploration est terminée.

3 Solvabilité

De même que dans le cas où l'agent ne peut pas attendre sur un site, un agent sans information sur les PV-graphes à explorer ne peut pas explorer tous les PV-graphes (même si on se limite aux PV-graphes étiquetés et hautement connexes).

Théorème 1 *Il existe une famille de PV-graphes étiquetés et hautement connexes tel qu'aucun agent ne peut explorer l'ensemble des PV-graphes de la famille s'il n'a pas d'information sur les PV-graphes qu'il explore.*

Élements de preuve. Intuitivement, la famille consiste en un petit PV-graphe G_0 et une infinité de PV-graphes "ressemblant" pendant un temps arbitrairement grand à G_0. L'agent doit entrer dans un état terminal en un temps fini t après avoir réaliser l'exploration de G_0. Il est possible de prouver qu'il existe un PV-graphe de la famille ne pouvant être différencié de G_0 par l'agent jusqu'au temps $t+1$ mais comportant un site de plus, qui ne sera donc jamais exploré par l'agent. □

4 Bornes inférieures

Due à la nouvelle capacité de l'agent, les bornes inférieures de [FMS09] ne s'appliquent pas dans notre modèle. Nous prouvons en fait dans cette section des bornes inférieures plus faibles, mais qui s'avèrent être optimales pour notre modèle (voir Section 5).

4.1 Borne inférieure sur le temps

Nous montrons ici une nouvelle borne inférieure sur la complexité en temps en $\Omega(np)$, également valable pour le cas restreint aux PV-graphes hautement connexes. Notons que le gain en complexité obtenu par rapport à la borne $\Theta(kp^2)$ lorsque l'agent ne peut pas attendre sur un site est d'autant plus important que la "densité" du PV-graphe est grande (un site est utilisé plusieurs fois dans une route et/ou par plusieurs transporteurs).

Théorème 2 *La complexité en temps du problème de l'exploration de PV-graphes est en $\Omega(np)$ dans le cas général comme dans le cas des PV-graphes hautement connexes. Ce résultat reste vrai même si l'agent connaît n, k et p, a une mémoire illimitée, et même dans le cas étiqueté.*

Élements de preuve. Les PV-graphes utilisés pour prouver ce théorème sont construits de la manière suivante. Les transporteurs sont numérotés de 1 à k. Les transporteurs de numéro impair, respectivement pair, sont de période p, respectivement $p-1$. Un transporteur i n'a de sites en commun qu'avec les transporteurs $i-1$ et $i+1$. (L'alternance des périodes assurent donc la haute connexité.) Plus précisément, le transporteur i passe une et une seule fois par un unique site visité par le transporteur $i-1$.

La preuve est ensuite basée sur le fait que l'agent ne sait pas précisément dans quel PV-graphe il se trouve. En particulier, l'agent ne sait pas sur quel site du transporteur $i-1$ ni à quels temps passera le transporteur suivant (numéro i). Il est possible de prouver, grossièrement, que l'agent doit attendre au moins $p-1$ unités de temps sur chaque site pour être sûr de trouver le transporteur suivant, d'où la borne obtenue. □

4.2 Borne inférieure sur le nombre de mouvements

Puisque l'agent a la possibilité d'attendre sur un site, la complexité en mouvements n'est plus nécessairement la même que la complexité en temps, comme c'était le cas dans le modèle de [FMS09]. Nous prouvons ici effectivement une borne sur le nombre de mouvements inférieure à la borne sur le temps.

Théorème 3 *La complexité en nombre de mouvements du problème de l'exploration de PV-graphes est en* $\Omega(\min\{kp, np, n^2\})$ *dans le cas général comme dans le cas des PV-graphes hautement connexes. Ce résultat reste vrai même si l'agent connaît n, k et p, a une mémoire illimitée, et même dans le cas étiqueté.*

Élements de preuve. Le principe général de la preuve est de construire une famille de PV-graphes dans lesquels la construction des routes obligent l'agent à visiter de nombreuses fois certains sites pour explorer les autres sites. Trois constructions différentes sont utilisées (suivant les valeurs relatives des différents paramètres) qui correspondent aux trois arguments du minimum. □

5 Notre algorithme

Notre algorithme utilise une borne supérieure B sur la période maximale p du PV-graphe (cf. Théorème 1). L'idée principale de l'algorithme consiste à descendre sur chaque site et, pendant $O(B)$ unités de temps, à noter tous les temps de passages de transporteurs sur ce site. Correctement gérées, ces informations permettent de cartographier le PV-graphe (liste des routes et des horaires de passage de tous les transporteurs).

Plusieurs précautions doivent être prises pour ne manquer aucun site et pour optimiser le nombre de mouvements. Par exemple, après chaque étude d'un site (pendant $O(B)$ unités de temps), l'algorithme calcule la plus petite période possible de chaque transporteur connu étant donné les informations déjà récoltés. Ceci permet de connaître tous les temps de passage futurs sur les sites étudiés. Afin d'éviter des déplacements inutiles, l'algorithme utilise la notion de transporteur courant. L'agent étudie tous les sites du transporteur courant avant de passer au suivant. Les identifiants des transporteurs rencontrés sont stockés sous la forme d'un arbre, un transporteur c' étant un fils d'un transporteur c si c' est découvert lorsque c est le transporteur courant. Les transporteurs sont traités en profondeur d'abord pour des raisons de performance.

Une analyse détaillée de l'algorithme complet permet de prouver le résultat suivant.

Théorème 4 *Un agent exécutant notre algorithme utilise au plus* $O(min\{kp, np, n^2\})$ *mouvements et* $O(nB)$ *unités de temps pour explorer et cartographier n'importe quel PV-graphe, où n est le nombre de sites, k le nombre de transporteurs et B une borne supérieure donnée à l'algorithme sur la plus grande période p.*

Notons que les algorithmes proposés dans [FMS09] ne sont optimaux en temps (et donc en mouvements) que si la borne supérieure sur p qu'ils utilisent est linéaire en p. Dans notre cas, d'après les Théorèmes 2 et 3, notre algorithme est de la même façon optimal en temps seulement si la borne supérieure B est linéaire en p, mais il est par contre toujours optimal en nombre de mouvements, aussi mauvaise que soit la borne B.

Références

[CFQS10] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. *arXiv CoRR*, abs/1012.0009, 2010.

[DP04] A. Dessmark and A. Pelc. Optimal graph exploration without good maps. *Theoretical Computer Science*, 326(1-3) :343–362, 2004.

[FMS09] P. Flocchini, B. Mans, and N. Santoro. Exploration of periodically varying graphs. In *20th Intl. Symp. on Algorithms and Computation (ISAAC)*, volume 5878 of *LNCS*, pages 534–543, 2009.

[IW11] D. Ilcinkas and A. M. Wade. On the power of waiting when exploring public transportation systems. In *15th Int. Conf. on Principles of Distributed Systems (OPODIS)*, volume 7109 of *LNCS*, pages 451–464, 2011.

[PP99] P. Panaite and A. Pelc. Exploring unknown undirected graphs. *Journal of Algorithms*, 33(2) :281–295, 1999.

[Rei05] O. Reingold. Undirected st-connectivity in log-space. *37th ACM Symposium on Theory of Computing (STOC)*, pages 376–385, 2005.